Express Mail Label No. EV303487038US

ERROR SIMULATION FOR A MEMORY MODULE

BACKGROUND

[0001] Computer systems, for example home computers or high-end computers operated as servers, may utilize memory to store data. The memory may comprise one or more memory modules that are connected together via a memory bus. The memory bus may utilize a signaling convention that facilitates the transfer of data between devices connected to the bus. The devices may comprise memory devices, data converters, and remote input/output ports.

[0002] To ensure that the memory associated with a computer system is functioning properly, an error detection mechanism within the computer system's chipset may monitor memory transactions and detect memory errors caused by hardware errors, for example stuck-at-1 faults (a data line stuck at a high voltage), stuck-at-0 faults (a data line stuck at a low voltage or grounded condition), and stuck-open faults (a data line driven neither high nor low, where the voltage on the data line therefore electrically floats).

[0003] To validate and test the error detection mechanism, hardware errors may be simulated on the data lines of a memory module. With memory bus speeds increasing, it may be difficult to reliably simulate hardware errors for the purpose of testing and validating the error detection mechanism.

SUMMARY

[0004] The problems noted above may be solved in large part by a memory module that simulates hardware errors. One exemplary embodiment may be a memory module that comprises a plurality of memory circuits, a plurality of data lines that transfer data to and from the memory circuits, and a switching device coupled to at least one of the plurality of data lines. The switching device

selectively operates to simulate a hardware error on the at least one of the plurality of data lines based on an input signal from a control logic external to the memory module.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] For a detailed description of exemplary embodiments of the invention, reference will now be made to the accompanying drawings in which:

[0006] Figure 1 illustrates a computer system constructed in accordance with embodiments of the invention;

[0007] Figure 2 illustrates a memory module constructed in accordance with embodiments of the invention;

[0008] Figure 3A illustrates a first exemplary configuration of the switching device of Figure 2 in accordance with embodiments of the invention;

[0009] Figure 3B illustrates a second exemplary configuration of the switching device of Figure 2 in accordance with embodiments of the invention;

[0010] Figure 4 illustrates a flow diagram of an error simulation procedure in accordance with embodiments of the invention; and

[0011] Figure 5 illustrates the logic of Figure 2 in accordance with embodiments of the invention.

NOTATION AND NOMENCLATURE

[0012] Certain terms are used throughout the following description and claims to refer to particular system components. As one skilled in the art will appreciate, computer companies may refer to a component by different names. This document does not intend to distinguish between components that differ in name but not function.

[0013] In the following discussion and in the claims, the terms "including" and "comprising" are used in an open-ended fashion, and thus should be interpreted to mean "including, but not limited to... ." Also, the verb "couple" or "couples" is intended to mean either an indirect or direct connection. Thus, if a first device couples to a second device, that connection may be through a direct connection, or through an indirect connection via other devices and connections.

DETAILED DESCRIPTION

[0014] The following discussion is directed to various embodiments of the invention. The embodiments disclosed should not be interpreted, or otherwise used, as limiting the scope of the disclosure unless otherwise specified. In addition, one skilled in the art will understand that the following description has broad application, and the discussion of any embodiment is meant only to be exemplary of that embodiment, and not intended to intimate that the scope of the disclosure is limited to that embodiment.

[0015] Figure 1 illustrates an exemplary computer system constructed in accordance with embodiments of the invention. System 100 may be any type of computer system, such as a laptop computer, a personal computer, or stand-alone computer operated as a server. The system 100 may comprise a single central processing unit (CPU) 102, as illustrated in Figure 1, or may comprise a plurality of CPUs arranged in a configuration where parallel computing may take place. The CPU 102 may couple to a memory controller 104 that manages a memory 106.

[0016] The memory 106 may comprise one or more memory slots, each slot designed to interface with a memory module, such as a dual inline memory module (DIMM). Although any number of memory slots may be used, eight memory slots 108 – 122 are illustrated in the memory 106. Each memory slot 108 – 122 may interface with a memory module (not specifically shown in Figure 1) that may comprise any type of memory circuit, such as double data rate (DDR), fast page mode (FPM), and extended data out (EDO). The memory slots 108 – 122 may be coupled together and to the memory controller 104 via a memory bus (not specifically shown) that facilitates the transfer of data between the memory slots 108 – 122 and the memory controller 104. In addition, control logic 124 may be coupled to the memory controller 104 and the memory 106. The control logic 124 may be a programmable logic array (PLA), a programmable logic device (PLD), or any other logic component capable of communicating with the memory 106 to simulate hardware errors.

[0017] Referring now to Figure 2, an exemplary memory module is shown in accordance with embodiments of the invention. The memory module 200 may

comprise one or more memory circuits that are capable of storing data. Although any number of memory circuits may be used, six memory circuits 202 – 212 are illustrated in the memory module 200.

[0018] Data may be transferred to and from the memory circuits 202 – 212 via a plurality of data lines. Each data line may couple to one or more input/output (I/O) pins 222 that interface the memory module 200 with one of the memory slot 108 – 122 (Figure 1). For example, a DIMM may possess 168 total I/O pins, 64 of which are coupled to distinct data lines. The pins not coupled to a data line may be used for various signals, such as address signals, power signals, ground signals, clock signals, and write strobe signals.

[0019] In accordance with embodiments of the invention, a switching device 214 may simulate a hardware error on one or more of the data lines in the memory module 200. The switching device 214 may attach to the memory module 200 at any suitable location. In accordance with embodiments of the invention, the attachment point may be selected such that the length of the wire or wires needed to couple the switching device 214 to the one or more data lines is minimized. By reducing the length of wire needed to couple the switching device 214 to the data lines, hardware errors may be more reliably simulated from the switching device 214. As shown in Figure 2, the switching device 214 is attached to an outer surface of one of the memory circuits 202-212, for example held in place on an outer surface of a packaged memory circuit by expoxy. The switching device 214 may be any type of switching device, for example IDT's QuickSwitch® QS3306A. The control logic 124 may couple to the switching device 214 via one or more enable lines 216.

[0020] Figure 3A illustrates the switching device 214 in accordance with at least some embodiments of the invention. The switching device 214 may comprise two switching units 302 and 304. Although mechanical switches are shown in Figure 3A, the switching units 302 and 304 may be any type of switching device, for example bipolar transistors or metal oxide semiconductor field effect transistors (MOSFETs). The switching unit 302 may couple to ground 308 through line 305, for example via one or more I/O pins 222 (Figure 2). The switching unit 304 may couple to a power source 306 through line 303, for

example via one or more I/O pins 222. The output signal of the switching device 214 may couple to the one or more data lines 310 in which a simulated hardware error is to take place. The enable lines 216 from the control logic 124 (Figure 2) may control operation of the switching units 302 and 304.

[0021] In the exemplary embodiments of Figure 3A, the state of the switching units 302 and 304 may be controlled by the combined states of the enable lines 216. When the switching unit 304 is conducting and the switching unit 302 is not conducting, the data line 310 may be driven to a high voltage level, representing a stuck-at-1 fault on the data line 310. When the switching unit 304 is not conducting and the switching unit 302 is conducting, the data line 310 may be driven to a low voltage level or a grounded condition, representing a stuck-at-0 fault on the data line 310. Thus, the switching device 214 as illustrated in Figure 3A is capable of selectively driving the data line 310 to a high and low voltage level.

[0022] Figure 3B illustrates alternative embodiments of the switching device 214. In the alternative embodiments, the switching device 214 may comprise three switching units 312, 314, and 316. Although mechanical switches are shown in Figure 3B, the switching units 312, 314, and 316 may be any type of switching device, for example bipolar transistors or MOSFETs. The switching unit 312 may couple to the ground 308, and the switching unit 314 may couple to the power source 306. The leads of the switching device 214 may electrically couple in series between the memory controller 104 and the memory circuit 202 to simulate the hardware errors in the memory module 200 (Figure 2). Thus, the embodiments of Figure 3B can simulate not only stuck-at-0 and stuck-at-1 hardware errors, but may also, by selective activation of the various switching units, simulate a stuck-open hardware error. The enable lines 216 from the control logic 124 (Figure 2) may control the switching units 312, 314, and 316.

[0023] Still referring to the embodiments illustrated by Figure 3B, when hardware errors are not being simulated, the switching unit 316 is conducting and the switching units 312 and 314 are not conducting. When the switching units 314 and 316 are conducting, and the switching unit 312 is not conducting, the data line 310 may be driven to a high voltage level, representing a stuck-at-1

hardware error. When the switching units 312 and 316 are conducting, and the switching unit 314 is not conducting, the data line 310 may be driven to a low voltage level, representing a stuck-at-0 hardware error. When the switching units 312, 314, and 316 are not conducting, the data line 310 may electrically float, representing a stuck-open hardware error. Thus, the switching device 214 is capable of driving the data line 310 to a high voltage level, driving the data line 310 a low voltage level or ground condition, and electrically floating the data line 310.

[0024] Figure 4 shows the relationship between an application 402, the control logic 124, and the switching device 214. An application 402 executed by the CPU 102 (Figure 1) may provide an interface to the control logic 124. The application 402 may utilize a basic input/output system (BIOS) call, a high-level driver, or any other means of accessing the control logic 124, such as an In-System Programming (ISP) header, to send a request 404 to the control logic 124. The request 404 may seek the simulation of one or more hardware errors, such as stuck-at faults and stuck-open faults. In addition, the request 404 may represent the location in embodiments where hardware errors may be simulated on multiple data lines (possibly by multiple switching devices 214) and duration of such hardware errors. Thus, the location may represent the data line or date lines on which to simulate a hardware error, and the duration may represent the amount of time the hardware error is to be simulated. The control logic 124 may receive the request 404 and responsive to the request 404 selectively assert the enable lines 216.

[0025] Referring now to Figure 5, an exemplary configuration of the control logic 124 is shown in accordance with at least some embodiments of the invention. The control logic 124 may comprise a serial data in (SDI) 502 line, a serial data out (SDO) 504 line, and a boundary scan (BSCAN) 508 line. The control logic 124 may be programmed by grounding (*i.e.,* setting to a low voltage value) the BSCAN 508 line and transferring data into the control logic 124 via the SDI 502 line. The request 404 (Figure 4) may be transferred to the control logic 124 via the SDI 502 line. Other inputs to the control logic 124, such as a reset 510 line and a clock 512 line, may be used. The control logic 124 may

couple to the switching device 214 of Figure 2 via the enable lines 216, as previously discussed. The enable lines 216 may control the switching device 214, which is responsible for simulating the hardware errors on the data lines of the memory module 200. In addition, one or more bus controllers, such as an inter-integrated circuits (I^2C) controller 514, that facilitate the transfer of data and commands from the application 302 (Figure 3) to the control logic 124 may also be associated with the control logic 124. For example, the control logic 124 may be coupled to and operating on an I^2C bus 524. The logic 214 may utilize a serial data (SDA) 516 line and a serial clock (SCL) 518 line to transfer data bi-directionally between the I^2C controller 514 and the I^2C bus 524. The request 404 may be transferred to the control logic 124 via the I^2C bus 524.

[0026] The control logic 124 may perform functions associated with hardware error simulation. For example, the control logic 124 may maintain a counter of the number of hardware errors to be simulated on the data lines of the memory module 200. In addition, the control logic may maintain a timer associated with the duration of the hardware errors to be simulated in the memory module 200.

[0027] The above discussion is meant to be illustrative of the principles and various embodiments of the present invention. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.